On Reasoning over Tracking Events

Daniel Rowe[1], Jordi Gonzàlez[2], Ivan Huerta[1], and Juan J. Villanueva[1]

[1] Computer Vision Centre / Computer Science Department, UAB, Barcelona, Spain
[2] Institut de Robòtica i Informàtica Industrial, UPC, Barcelona, Spain

Abstract. High-level understanding of motion events is an essential task in any system which aims to analyse a human-populated scene. In this paper, a principled event-management framework is proposed, and it is included in a hierarchical and modular tracking architecture. Multiple-target interaction events, and a proper scheme for tracker instantiation and removal according to scene events, are considered. Multiple-target group management allows the system to switch among different operation modes. Robust and accurate tracking results have been obtained in both indoor and outdoor scenarios, without considering a-priori knowledge about either the scene or the targets.

1 Introduction

High-level event understanding is an essential task of any instance of a generic *Image-sequence Evaluation* (ISE) system [9], in particular HSE systems [3]. These transform image-sequence data —recorded in human-populated scenes— into semantic descriptions; subsequently, these descriptions are processed, and the system reacts in terms of signal triggers or conceptual terms. Such a system could perform a smart video surveillance, an intelligent gestural user-computer interfaces, or any other application in orthopedics, sports, natural-language scene description, or computer-animation fields [1,5,7].

A robust and accurate multiple-people tracking is a crucial component of any HSE system. However, a proper event detection and management is critical for tracking success. Further, this provides a valuable knowledge to achieve scene understanding. Thus, event management requires (i) considering simultaneously multiple target interactions, specially when no assumption is made with respect to the targets' trajectories; and (ii), since in every open-world scenario, targets can enter and exit the scene, a procedure has to be implemented to reliably perform tracker instantiation and removal.

Despite this interest and the increasing number of proposed algorithms which deal with multiple interacting targets in open-world scenarios, this still constitutes an open problem which is far from been solved. Yang et al. [13] proposed a system with some similarities to ours, albeit no filtering is carried out, grouped targets are not independently tracked, and the cues and models used are essentially different. Wu et al. [12] address occlusions events within a Particle Filter (PF) framework by implementing a Dynamic Bayesian Network (DBN) with an extra hidden process for occlusion handling. BraMBLe [6] is an interesting approach to multiple-blob tracking which models both background and foreground

using Mixtures of Gaussians (MoG). However, no model update is performed, there is a common foreground model for all targets, and suffers for the curse of dimensionality, as all PF-based methods which tackle multiple-target tracking combining information about all targets in every sample. Alternatively, several approaches take advantage of 3D information by making use of a known camera model and assuming that agents move on a known ground plane. These and other assumptions relative to a known Sun position or constrained standing postures allow the system presented in [14] to initialise trackers on people who do not enter the scene isolated.

Only recently simultaneous tracking of numerous target has been considered. This force tracking systems to consider complex interacting events. In this paper, a principled event-management framework is proposed, and included in a hierarchical and modular tracking architecture. Multiple-target interaction events are handled by means of a state machine, which consider all possible grouping configuration. This will crucial in order to achieve successful performances, by allowing the system to switch among different tracking approaches [8]. Further, a proper scheme for tracker instantiation and removal is proposed, which is basic in any open-world application.

The remainder of this paper is organized as follows. Section 2 outlines the system architecture. Section 3 details the event management approach. Section 4 shows some experimental results obtained from well-known databases, and finally, section 5 summarises the conclusions, and proposes future-work lines.

2 Tracking Framework

Due to the inherent complexity involved in non-supervised multiple-human tracking, a structured framework is proposed to accomplish this task. We take advantage of the modular and hierarchically-organised system published in preliminary works [4,11]. This is based on a set of co-operating modules distributed in three levels. These are defined according to the different functionalities to be performed, namely target detection, low-level tracking, and high-level tracking, see Fig. 1. A remarkable characteristic of this architecture is that the tracking task is split into two levels: a lower level based on a short-term blob tracker, and a long-term high-level appearance tracker. The latter automatically builds and tunes multiple appearance models, manages the events in which the target is involved, and selects the most appropriate tracking approach according to these.

In general, reliable target segmentation is critical in order to achieve an accurate feature extraction without considering any prior knowledge about potential targets, specially in dynamic scenes. However, complex interacting agents who move through cluttered environments require high-level reasoning. Thus, this proposal combines in a principled architecture both bottom-up and top-down approaches: the former provides the system with initialisation, error-recovering and simultaneous modelling and tracking capabilities, while the latter builds the models according to a high-level event interpretation, and allows the system to switch among different operation modes.

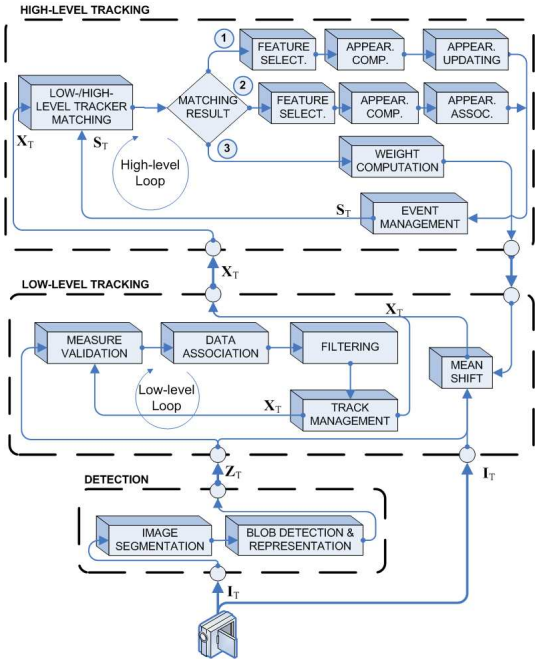

Fig. 1. System architecture. $\mathbf{I}_t$ represents the current frame, $\mathbf{Z}_t$ represents the observations, $\mathbf{X}_t$ the target's low-level state, and $\mathbf{S}_t$ the target's high level state. Matching results are explained in the text.

The lower level performs target detection. First, the segmentation task is accomplished following a statistical colour background-subtraction approach. Next, the obtained image masks are filtered, and object blobs are extracted. Each blob is labelled, their contours are computed, and they are parametrically represented. Consequently, the spurious structural changes that they may undergo are constrained. These include target fragmentation due to camouflage, or the inclusion of shadows and reflections. Moreover, this representation can be handled by the low-level tracker, thereby filtering the target state and reducing also these effects. An ellipse representation —which keeps the blob first and second order moments— is chosen [2,10]. Thus, the j-observed blob at time t is given by the vector $\mathbf{z}_j^t = \left(x_j^t, y_j^t, h_j^t, w_j^t, \theta_j^t\right)$, where x_j^t, y_j^t represent the ellipse centroid, h_j^t, w_j^t are the major and minor axes, respectively, and the θ_j^t gives the angle between the abscissa axis and the ellipse major one. Low-level trackers establish coherent target relations between frames by setting correspondences between observations and trackers, and by estimating new target states according to the associated observations using a bank of Kalman filters. Finally, the *track-management* module (i) initiates tentative tracks for those observations which are not associated; (ii) confirms tracks with enough supporting observations; and (iii) removes low-quality ones. Results are forwarded to high-level trackers, and fed back to the measure-validation module. See [4] for details.

A high-level tracker is instantiated whenever a low-level track is first confirmed. Hence, tracking events can be managed. This allows target tracking even when image segmentation is not feasible, and low-level trackers are removed, such as during long-duration occlusions or grouping. As a result of the tracker matching, three cases are considered: (i) if the track is stable, the target appearance is computed and updated, see matching result (1) in Fig. 1; (ii) those high-level trackers which remain orphans are processed to obtain an appearance-based data association, thereby establishing correspondences between lost high-level trackers and new ones, see matching result (2). The details of this procedure can be found in [11]; and, (iii) those targets which have no correspondence are tracked in a top-down process using appearance-based trackers, see matching result (3). An *event* module determines what is happening within the scene, such as target grouping or entering the scene. These results are fed back, thereby allowing low-level and high-level tracker matching.

3 Event Management

Multiple-people tracking requires considering potential target interactions, specially when no assumption is made with respect to the targets' trajectories. These interactions will be referred in the following as *interaction events*. Further, in every open-world scenario targets can enter and exit the scene, or a Region Of Interest (ROI) defined on it. These events will be referred as *scene events*, and they have an important role in matching low-level and high-level trackers, and in managing the latters. Both types of events will be managed as follows.

3.1 On interaction events

A proper detection of interaction events is crucial to achieve successful performances, since a different tracking approach must be used in each case. On the one hand, whenever a detected blob clusters more than one target, tracking by motion detection is no longer feasible, and no accurate target position can be obtained. On the other hand, appearance-based trackers, like those based on mean-shift methods, suffer from a poor target localisation, and therefore they are not the optimal choice when an appropriate detection can be performed. Thus, by detecting these events, several operation modes could be introduced and properly selected. Further, this represents a significant knowledge which can be used for scene understanding.

Two targets are said to be *in-collision* when their *safety areas* superpose themselves. These areas are defined according to the targets' sizes. Thus, the following states are defined: (i) a target is considered as *single* if it does not collide with any other target within the scene; (ii) targets are said to be *grouping* if they do collide, but no group is being tracked in their area; (iii) targets are considered as *grouped* if they collide, they are over a group tracker area, and the group tracker is currently associated with an observation; (iv) finally, trackers are said to be *splitting* once the group has no longer an observation, but they

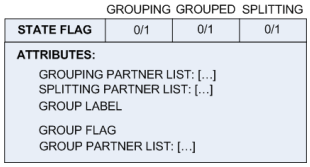

Fig. 2. Target state coding.

do still collide. The frame rate is supposed to be high enough so that a target cannot change from grouped to single without ever being splitting.

Unfortunately, the above-presented classification does not suffice in complex scenarios where clusters of more than one target may be formed; for instance, one target could be grouping with a second one at the same time as splitting from a third one. Hence, the aforementioned scheme should be generalised by taking into account multiple and different target interactions.

The interaction state is coded using a three-bit vector, where each bit point outs whether the target is grouping, grouped or splitting. When every bit is set to zero, the target's state is single. Otherwise, the state could be a mixture of the previously defined situations. Secondly, several attributes are associated with each state. These point out relevant information to solve queries about current interaction events: which targets are interacting, which ones are simultaneously grouping and splitting, with which targets are they grouping, etc. Two cases are distinguished, depending on whether the tracker tracks a target or a group of them. In the first case, two lists of grouping and splitting partners are kept. Further, the group label, if this exists, is stored. In the second one, a flag pointing out that the tracker tracks a group is defined. In addition, a list of grouped targets is also kept. Thus, the eight possible states include all potential tracking situation, and these, along with the associated attributes, constitute all the necessary knowledge to solve any query relative to target interaction, see Fig 2.

Next, several events must be taken into account in order to define state transitions. These include issues such as target collision with another target, or with a group, whether the group has an associated observation or not, if there are new partners in collision, or whether old ones are no longer partners.

Thus, once all targets' positions and sizes are estimated, a collision map is computed. The collision map is also used to determine whether a new-born tracker represents a group: in this case, it is instantiated over a collision zone. Then, when two single targets are colliding, and none of them is a new target, their states change into grouping. If they also collide with a group tracker with an associated observation, their states are set to grouped. Once the group tracker has no longer an associated observation, but they still collide, their states change into splitting. More complex situations can be taken into account by considering the previous and current partner list. Finally, a tracker that stop colliding at any state becomes single again. As an example of complex interaction, consider a target whose state is grouped; then, the following events take place: (i) it is colliding with some other targets, (ii) the group has no associated observation,

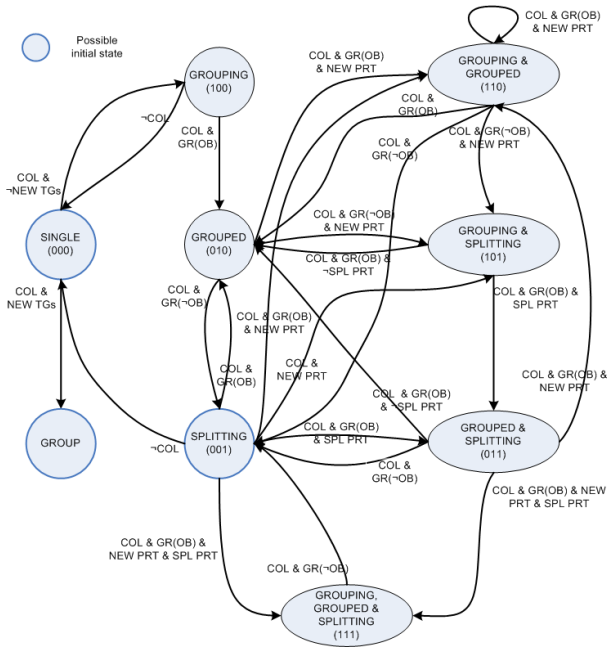

Fig. 3. Group management. Eight possible target states, and a state for group trackers, are defined (represented by circles). Interaction events are denoted by arrows.

and (ii) new partners are also colliding. As a result, it changes its state into *grouping and splitting*.

The state machine that models the group management is defined by eight plus one states. The formers are defined for target trackers, and the latter for group trackers. Thus, there are 56 potential transitions between target states, although a fraction of them are not feasible according to the aforementioned assumptions. For instance, grouped targets cannot become single, since they have to split before. There is also possible to perform changes in the attributes without this meaning a state transition. This is the case when several targets are already grouping, and a new one joins them. The state machine is show in Fig. 3. Notice that some of the less frequent transitions are not drawn for the sake of clarity. It should be also remarked that it is not possible to add new partners to a group without first removing the group and then creating a new one. This happens because new observations won't be assigned to the former group since both position are shape would have undergone important changes. This is however a desirable effect since the new group would have a different number of partners, and therefore it is actually a different group.

Although the current proposal do not allow yet to initially track people who do not enter into the scene isolated, it do detect them as they split and stable trackers are instantiated over the group region.

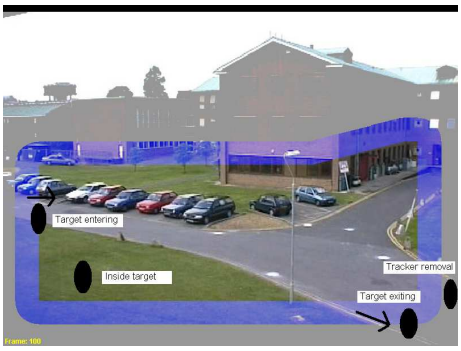

Fig. 4. Scene regions. The three regions defined on an image from PETS database.

3.2 On Scene Events

A proper handling of scene events is essential in order to achieve successful system performances in open-world applications. In these, the number of targets within the scene is not a-priori known, and it may vary as new targets enter the scene, or other ones exit it. By defining a Region of Interest (ROI) within the scene boundaries three aims can be achieve: (i) it is not necessary to fully process the whole image, and therefore this favours accomplishing real-time performances; (ii) the number of false positives can be effectively reduced, by avoiding detections in non-plausible or non-interesting areas, like the sky in a pedestrian-surveillance application; and (iii) targets can be completely segmented.

Three regions are defined: a ROI, a security border, and non-interesting areas. These are used to define where targets can be detected, where low-level and high-level trackers can be instantiated, and when they can be removed. The security border prevents the system from creating and removing trackers following the same target placed on the ROI frontier.

Thus, pixel segmentation is carried out in the whole image, since targets' sizes are not a-priori known. However, targets are only detected if the centroid of the corresponding blob lies within the ROI or the security border. For each detected target, a low-level tracker is instantiated. Once a low-level tracker is confirmed, a high-level tracker can be instantiated. This requires that the tracker has an associated observation, which implies that the target centroid is within the aforementioned area, and that the target is at least partially within the ROI. High-level trackers are instantiated as *entering*, except when they come from a group that have split. This status last until they completely lie within the ROI. When a part of the target is partially outside the ROI and the security border, the target is marked as *exiting*. The target can now either return to the ROI, or lie completely outside the area defined by the ROI and the security border. The latter implies the tracker removal. Trackers are also removed if they are partially in the outer zone and they are being tracked by a low-confidence appearance tracker, thereby avoiding a senseless gradient-based search when the target has actually exited. An example is shown in Fig. 4.

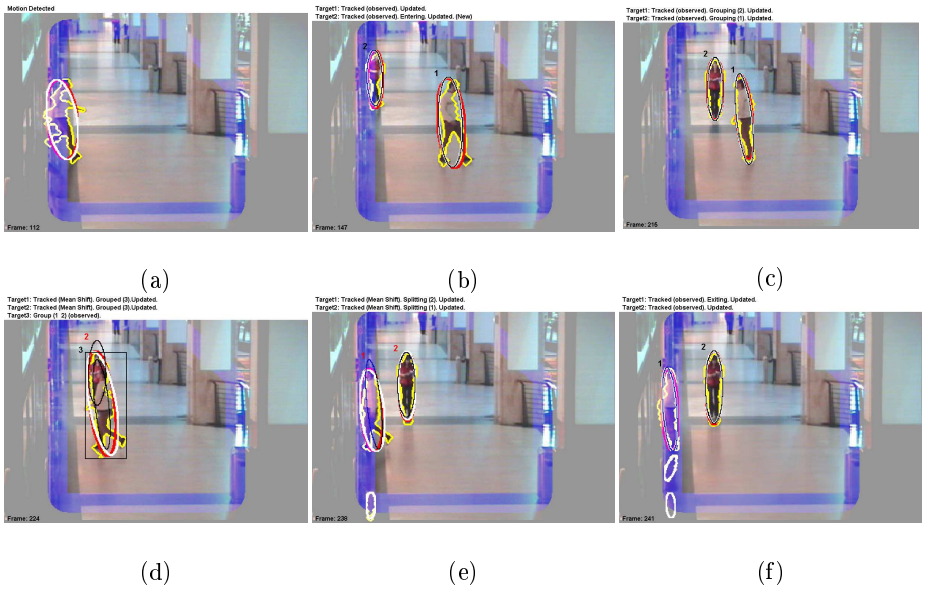

Fig. 5. Tracking results on a indoor sequence.

4 Experimental Results

The performance of the system has been tested using sequences taken from two well-known data-sets: the CAVIAR database[1], and PETS 2001 Test Case Scenario[2]. The former corresponds to indoor sequences which have been recorded in a mall centre, whereas the latter contains outdoor sequences taken in a scene which includes roads, parking places, green areas, and several buildings.

In the sequence *OneLeaveShopReenter1cor* (CAVIAR database, 389 frames at 25 fps, 384 x 288 pixels), two targets are tracked simultaneously, despite their being articulated and deformable objects whose dynamics are highly non-linear, and that move through an environment which locally mimics the target colour appearance. The first target performs a rotation and heads towards the second one, eventually occluding it. The background colour distribution is so similar to the target one that it constitutes a strong source of clutter. Furthermore, several oriented lighting sources are present, dramatically affecting the target appearance depending on its position and orientation (notice the bluish effect on the floor on the right of the corridor, and the reddish one on the floor of the left of the corridor). Thus, significant speed, size, shape and appearance changes can be observed, jointly with events such as people grouping, partial occlusions and group splitting.

[1] http://homepages.inf.ed.ac.uk/rbf/CAVIAR
[2] http://peipa.essex.ac.uk/ipa/pix/pets

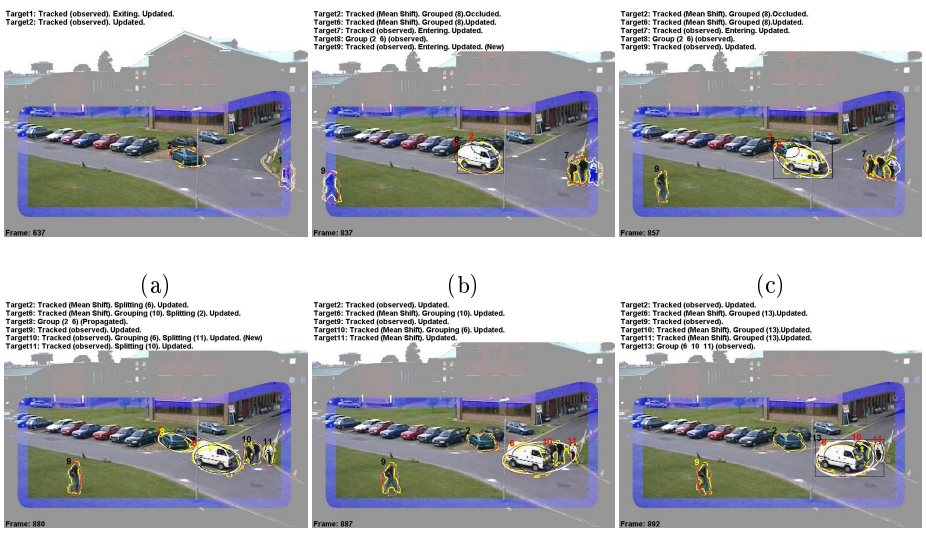

Fig. 6. Tracking results on an outdoor sequence.

The sequence *DATASET1_TESTING_CAMERA1* (PETS database, 2688 frames at 29.97 fps, 768 x 576 pixels) presents a high variety of targets entering into the scene: three isolated people, two groups of people, three cars, and a person who exits from a parked car. These cause multiple tracking events in which several targets are involved in different grouping, grouped, and splitting situations simultaneously.

Targets are accurately tracked along both sequences. All events are correctly detected. Fig. 5 shows a sequence successful event detections for both targets. Blobs in motion are detected and low-level trackers are created. Once they enter the scene, high-level trackers are instantiated and associated to the stable low-level ones. A grouping event is correctly detected, making the operation mode change into appearance tracking. Despite the strong occlusion of target 2, both targets are accurately tracked while they are grouped. Finally, the split event is detected and the operation mode is again changed into tracking by motion. Fig. 6 shows a more complex sequence of interaction events. A group enter the scene together, see Fig. 6.(b), but an independent tracker have been associated to one person as they momentarily split. In Fig. 6.(d) targets 2 and 6 are tracked using appearance-based methods, while targets 9,10 and 11 are tracked by motion detection. In this frame, target 2 is splitting from 6, which is also grouping with target 10. The latter is in fact a group of two people who are grouping with target 6 while splitting from target 11. In Fig. 6.(f), targets 6,10 and 11 have conformed a stable group and all of then are being tracked by means of appearance tracking.

5 Concluding Remarks

A structured multiple-target tracking framework is presented. No a priori knowledge about either the scene or the targets is required. A remarkable characteristic of the system is its ability to manage multiple interactions among several targets. This provides a valuable knowledge in order to obtain high-level scene descriptions, while allowing the system to switch among different operation modes. The latter is crucial to achieve successful performances: non-supervised multiple-human tracking is a complex task which demands different approaches according to different situations.

Experiments on complex indoor and outdoor scenarios have been successfully carried out, thereby demonstrating the system ability to deal with difficult situations in unconstrained and dynamic scenes. Future work will focus on segmenting groups of people who do not enter the scene isolated, thereby allowing a robust and independent target tracking. In addition, targets will be classified by distinguishing among people, vehicles and other objects in motion.

References

1. R. Collins, A. Lipton, and T. Kanade. A System for Video Surveillance and Monitoring. In *8th ITMRRS, Pittsburgh, USA*, pages 1–15. ANS, 1999.
2. D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based Object Tracking. *PAMI*, 25(5):564–577, 2003.
3. J. Gonzàlez. *Human Sequence Evaluation: The Key-frame Approach.* PhD thesis, UAB, Spain, 2004.
4. J. Gonzàlez, D. Rowe, J. Andrade, and J.J. Villanueva. Efficient Management of Multiple Agent Tracking through Observation Handling. In *6th VIIP, Mallorca, Spain*, pages 585–590. IASTED/ACTA PRESS, 2006.
5. I. Haritaoglu, D. Harwood, and L. Davis. W4: real-time surveillance of people and their activities. *PAMI*, 22(8):809–830, 2000.
6. M. Isard and J. MacCormick. BraMBLe: A Bayesian Multiple-Blob Tracker. In *8th ICCV, Vancouver, Canada*, volume 2, pages 34–41. IEEE, 2001.
7. R. Kahn, M. Swain, P. Prokopowicz, and R. Firby. Gesture Recognition Using the Perseus Architecture. In *CVPR, San Francisco, USA*, pages 734–741. IEEE, 1996.
8. T. Matsuyama and V. Hwang. *SIGMA A Knowledge Based Aerial Image Understanding System.* Plenum Press, 1990.
9. H. Nagel. Image Sequence Evaluation: 30 years and still going strong. In *15th ICPR, Barcelona, Spain*, volume 1, pages 149–158. IEEE, 2000.
10. K. Nummiaro, E. Koller-Meier, and L. Van Gool. An Adaptive Color-Based Particle Filter. *IVC*, 21(1):99–110, 2003.
11. D. Rowe, I. Reid, J. Gonzàlez, and J. Villanueva. Unconstrained Multiple-people Tracking. In *28th DAGM, Berlin, Germany*, pages 505–514. Springer, 2006.
12. Y. Wu, T. Yu, and G. Hua. Tracking Appearances with Occlusions. In *CVPR, Wisconsin, USA*, volume 1, pages 789–795. IEEE, 2003.
13. T. Yang, S. Li, Q. Pan, and J. Li. Real-time Multiple Object Tracking with Occlusion Handling in Dynamic Scenes. In *CVPR, San Diego, USA*, volume 1, pages 970–975. IEEE, 2005.
14. T. Zhao and R. Nevatia. Tracking Multiple Humans in Complex Situations. *PAMI*, 26(9):1208–1221, 2004.